Mobile Robotics: Assignment 3 Report

Mohsin Mustafa - **20161131**, Shyamgopal Karthik - **20161221**

November 13, 2019

Contents

1 Bi-Quadratic Polynomial

1.1 Formulation

We consider 2 4th degree polynomials: $x(t) = A_0t^4 + A_1t^3 + A_2t^2 + A_3t + A_4$ and $y(t) = B_0t^4 + B_1t^3 + B_2t^2 + B_3t + B_4$ The constraints that are imposed are:

1. Position at $t = 0$ must be the starting position
2. Position at $t = 5$ must be the final position
3. Velocity at $t = 0$ must be 0.
4. Velocity at $t = 5$ must be 0.
5. At an arbitrary item $t = t_{in}$ position must be at the given intermediate position. We set $t_{im} = 1$

This gives us the constraint matrix

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 625 & 125 & 25 & 5 & 1 \\ 500 & 75 & 10 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}, b_x = \begin{bmatrix} 3 \\ 0 \\ 9 \\ 0 \\ 1 \end{bmatrix}, b_y = \begin{bmatrix} 0 \\ 0 \\ 5 \\ 0 \\ 2.5 \end{bmatrix}$$

The co-efficients for $x(t)$ and $y(t)$ are recovered as: $x = (A^TA)^{-1}A^Tb_x$ $y = (A^TA)^{-1}A^Tb_y$

1.2 Results

The results are shown in Figure 1, Figure 2 and Figure 3.

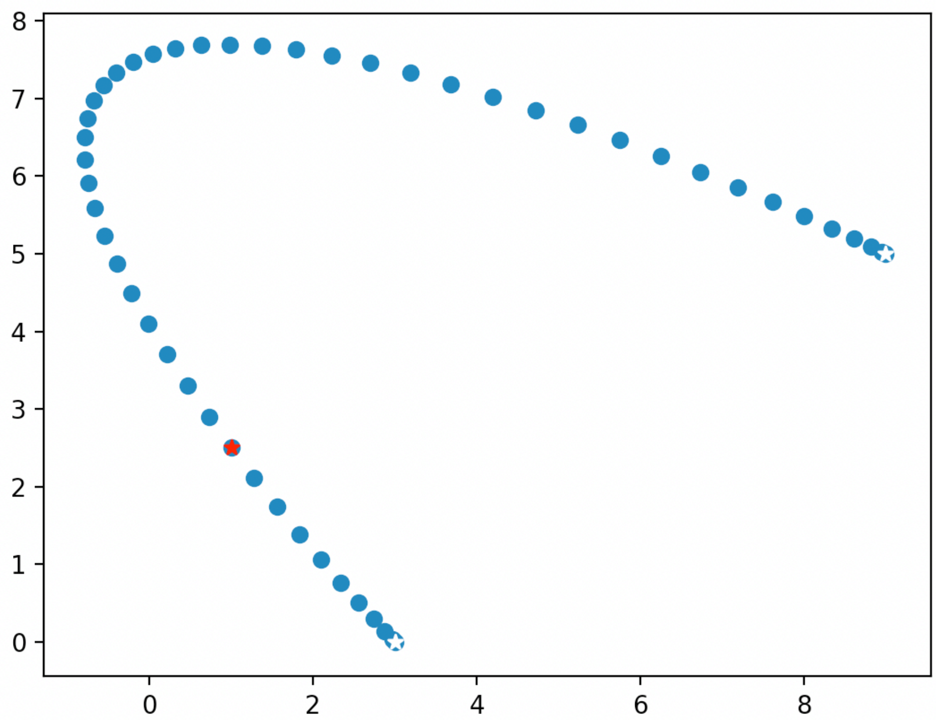

Figure 1: Plot showing the path taken by the robot for the biquadratic polynomial

---

2 Bernstein Polynomial

2.1 Formulation

The Bernstein Polynomials can be defined here as:

$$x(t) = \sum_{i=0}^{5} C_i^5 \tau^i (1-\tau)^{5-i} X_i$$

$$y(t) = \sum_{i=0}^{5} C_i^5 \tau^i (1-\tau)^{5-i} Y_i$$

Applying the same constraints as before, we get the constraint matrix A as:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0.32768 & 0.4096 & 0.2048 & 0.0512 & 0.0064 & 0.00032 \end{bmatrix}$$

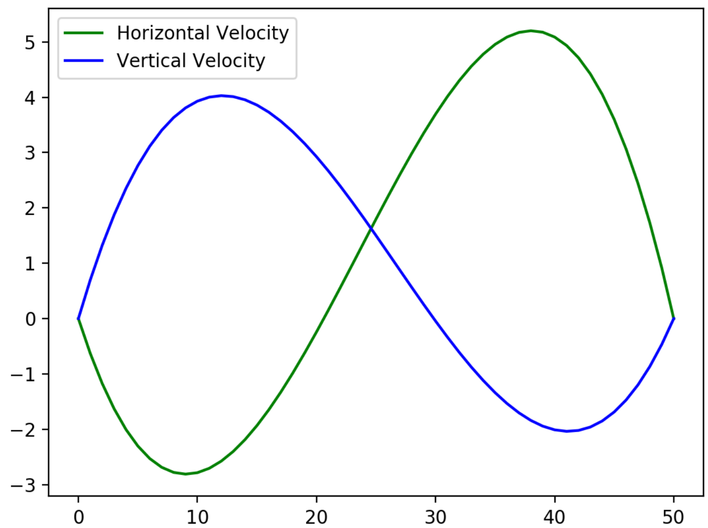

Figure 2: Plot showing the velocity of the robot for the biquadratic polynomial

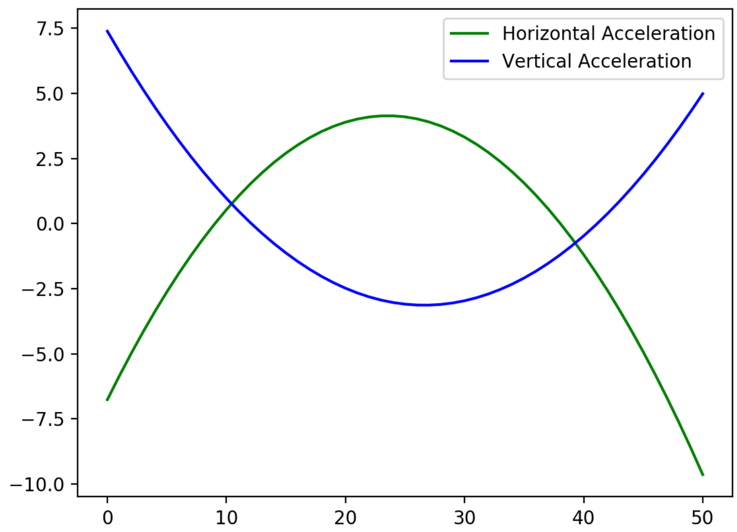

Figure 3: Plot showing the acceleration(horizontal and vertical) of the robot for the biquadratic polynomial

The vectors b_x and b_y are as follows:

$$b_x = \begin{bmatrix} 3 \\ 0 \\ 9 \\ 0 \\ 1 \end{bmatrix}, b_y = \begin{bmatrix} 0 \\ 0 \\ 5 \\ 0 \\ 2.5 \end{bmatrix}$$

The co-efficients for $x(t)$ and $y(t)$ can be obtained as:

$$x = (A^T A)^{-1} A^T b_x$$

$$y = (A^T A)^{-1} A^T b_y$$

Velocity and acceleration (across x-axis) are calculated as follows:

$$v_x(t) = x'(t) = \sum_{i=0}^{5} \frac{C_i^5 X_i}{t_f - t_i} \left[i\tau^{i-1}(1-\tau)^{n-i} - (n-i)\tau^i(1-\tau)^{n-i-1} \right]$$

$$a_x(t) = v_x'(t) = \sum_{i=0}^{5} \frac{C_i^5 X_i}{(t_f - t_i)^2} \times [i(i-1)\tau^{i-2}(1-\tau)^{n-i} \\ - 2i(n-i)\tau^{i-1}(1-\tau)^{n-i-1} + (n-i)(n-i-1)\tau^i(1-\tau)^{n-i-2}]$$

2.2 Results

The results are shown in Figure 4, Figure 5 and Figure 6.

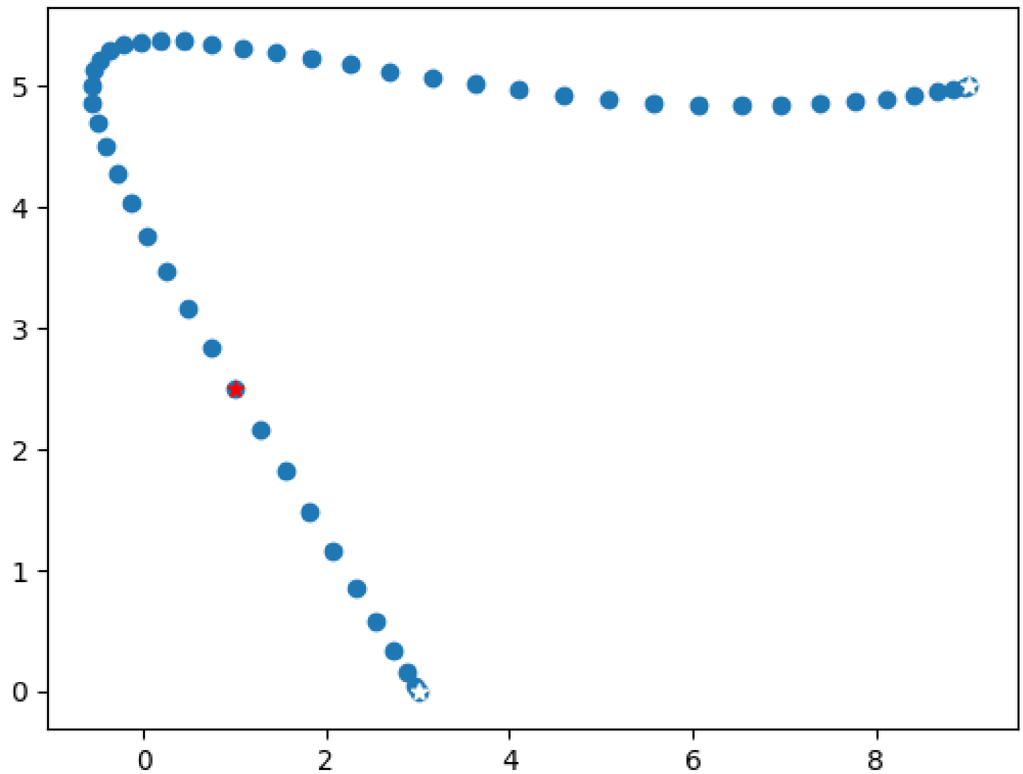

Figure 4: Plot showing the path taken by the robot for the Bernstein Polynomial

---

3 Obstacle Avoidance

3.1 Algorithm

The crux behind the algorithm followed for avoiding an obstacle (specifically a circular obstacle of radius r centered at (x_c, y_c)) is to choose alternate points to intersect such that the path will steer away from the obstacle.

First a path is made that starts from **Start - Intersection Point - Obstacle Center - End**. Using this path, we find the perpendicular vector at (x_c, y_c). The perpendicular is used to find increasing offset points (o_{xi}, o_{yi}). Each iteration until we get a successful path, the offset points move away from the obstacle center along the perpendicular. In the i^{th} iteration, we create a new path **Start - Intersection Point - i^{th} Offset Point - End**. If this path does *not* intersect the circle, then we have found a valid path. Else we move onto the next iteration.

The path is found in each case using the algorithm used and discussed in Q1. A step-by-step working example (of the provided points) is visualized in Figure 7.

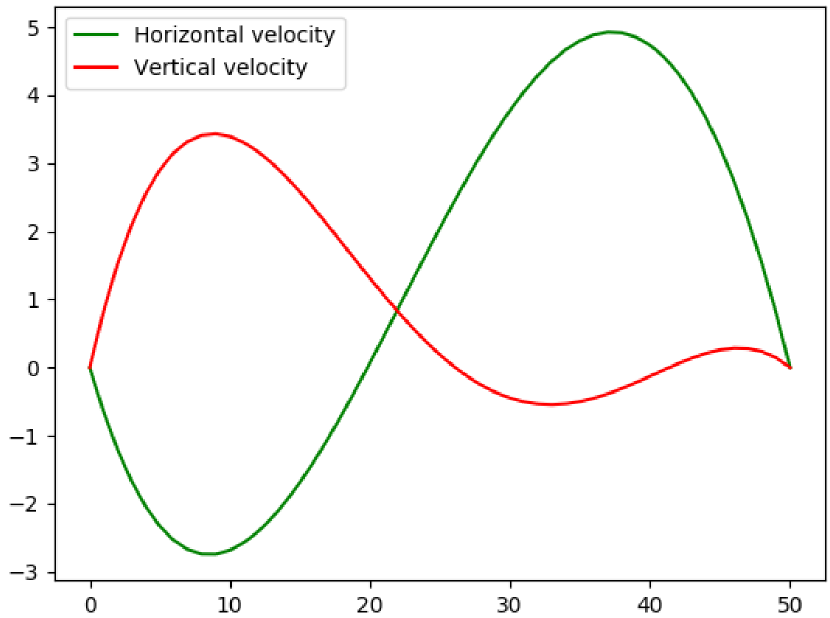

Figure 5: Plot showing the velocity for the Bernstein Polynomial

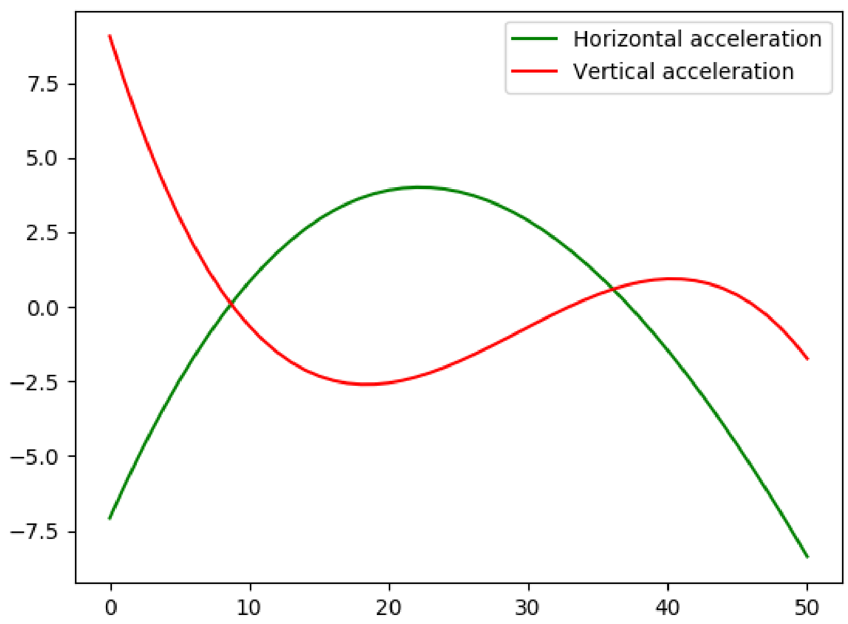

Figure 6: Plot showing the Acceleration for the Bernstein Polynomial

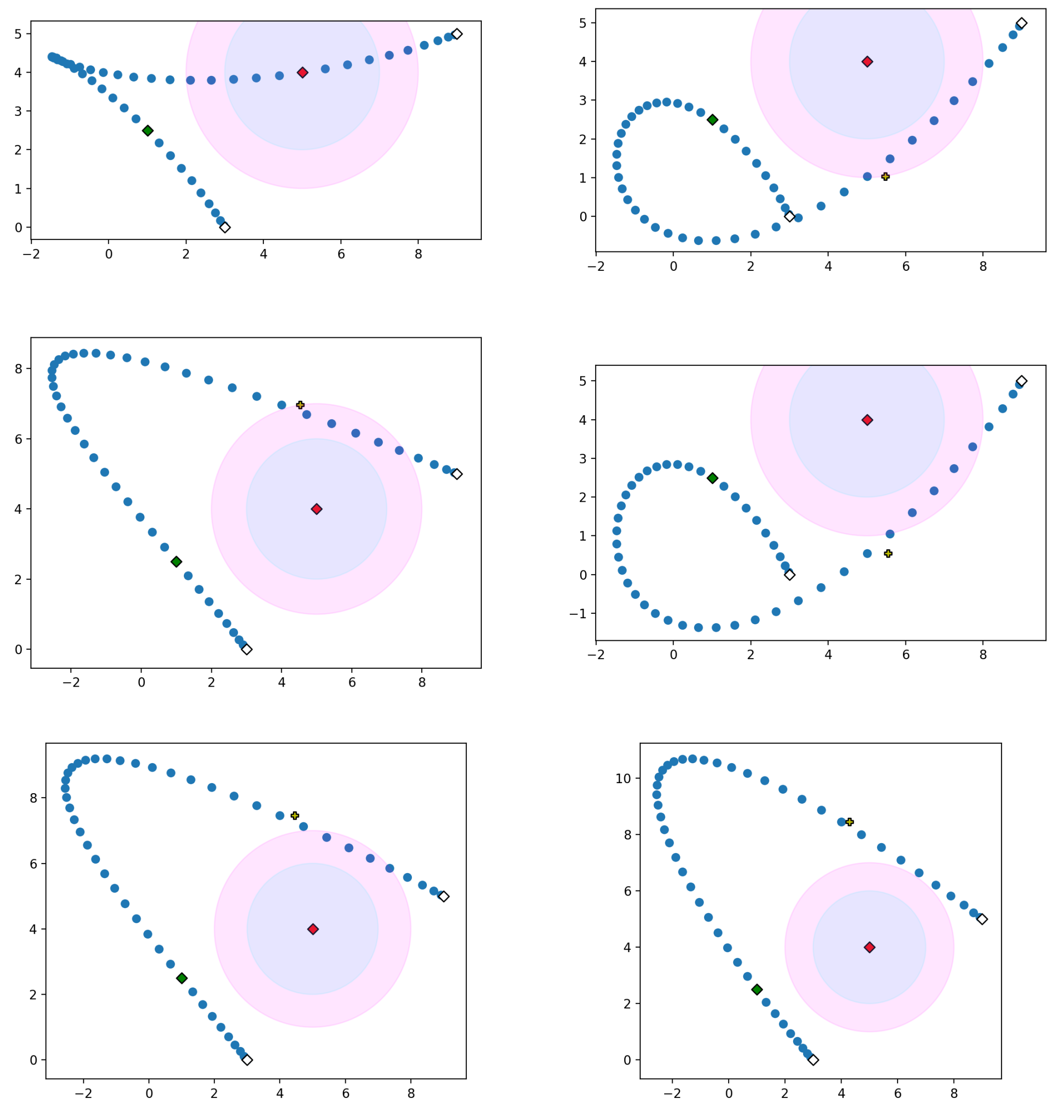

Figure 7: The top-left image shows the original path computed to find the perpendicular to the center. From the second image onwards the yellow cross signifies the offset point of that particular iteration. Notice how it moves away from the center each two iterations, on either side of the center. The blue ring is the obstacle, and the pink circle shows the boundary that the center of our robot must not encroach to prevent collision. The green square is the intersection point, and the white squares are the beginning and ending points. The values are scattered for each dt time. (The second-last iteration was removed from this image due to lack of space.)